23-2208

United States Court of Appeals for the Federal Circuit

FINTIV, INC.,

Plaintiff-Appellant

v.

APPLE INC.,

Defendant-Appellees

Appeal from the United States District Court for the Western District of Texas, Case No. 1:21-cv-00896-ADA, District Judge Alan D. Albright

APPELLANT FINTIV, INC.'S CORRECTED NON-CONFIDENTIAL REPLY BRIEF

Meredith Martin Addy
Charles A. Pannell, III
ADDYHART P.C.
10 Glenlake Parkway, Suite 130
Atlanta, Georgia 30328
312.320.4200
meredith@addyhart.com
cpannell@addyhart.com

Caren A. Yusem
ADDYHART P.C.
1101 Pennsylvania Avenue, N.W.
Suite 300
Washington, DC 20004
312.804.4885
caren@addyhart.com

Jonathan K. Waldrop
Darcy L. Jones
Marcus A. Barber
John W. Downing
Heather S. Kim
ThucMinh Nguyen
Kasowitz Benson Torres LLP
333 Twin Dolphin Drive, Suite 200
Redwood Shores, CA 94065
650.453.5170
jwaldrop@kasowitz.com
djones@kasowitz.com
mbarber@kasowitz.com
jdowning@kasowitz.com
hkim@kasowitz.com
tnguyen@kasowtiz.com

Attorneys for Appellant Fintiv, Inc.

(continued on next page)

Paul G. Williams
Kasowitz Benson Torres LLP
1230 Peachtree Street, NE
Suite 2445
Atlanta, GA 30309
404.260.6102
pwilliams@kasowitz.com

Attorneys for Appellant Fintiv, Inc.

(continued from previous page)

February 29, 2024

CERTIFICATE OF INTEREST

Counsel for Appellant Fintiv, Inc., certifies:

1. **Represented Entities.** Provide the full names of all entities represented by undersigned counsel in this case. Fed. Cir. R. 47.4(a)(1).

   Fintiv, Inc.

2. **Real Party in Interest.** Provide the full names of all real parties in interest for the entities. Do not list the real parties if they are the same as the entities. Fed. Cir. R. 47.4(a)(2).

   None/Not Applicable.

3. **Parent Corporations and Stockholders.** Provide the full names of all parent corporations for the entities and all publicly held companies that own 10 percent or more of the stock of the entities:

   None/Not Applicable.

4. **Legal Representatives.** List all law firms, partners, and associates that (a) appeared for the entities in the originating court or agency or (b) are expected to appear in this court for the entities. Do not include those who have already entered an appearance in this court. Fed. Cir. 4. 47.4(a)(4).

   (a) Marc E. Kasowitz, Kasowitz Benson Torres LLP
   Jeceaca An, Kasowitz Benson Torres LLP
   Julianne Laporte, Kasowitz Benson Torres LLP
   Chen Jia, Kasowitz Benson Torres LLP

   Jack Shaw, formerly Kasowitz Benson Torres LLP
   Daniel C. Miller, formerly Kasowitz Benson Torres LLP
   Rodney R. Miller, formerly Kasowitz Benson Torres LLP
   Gurtej Singh, formerly Kasowitz Benson Torres LLP
   Shelley Ivan, formerly Kasowitz Benson Torres LLP
   Trevor J. Welch, formerly Kasowitz Benson Torres LLP

Raymond W. Mort, III, The Mort Law Firm, PLLC

J. Mark Mann, Mann, Tindel & Thompson
Andy W. Tindel, Mann, Tindel & Thompson
G. Blake Thompson, Mann, Tindel & Thompson

Craig D. Cherry, Cherry Johnson Siegmund James PLLC
Justin W. Allen, Cherry Johnson Siegmund James PLLC

George Philip Cowden, The Cosden Law Firm PLLC

(b) Not applicable.

5. **Related Cases.** Provide the case titles and numbers of any case known to be pending in this court or any other court or agency that will directly affect or be directly affected by this court's decision in the pending appeal. *See* Fed. Cir. R. 47.4(a)(5) and 47.5(b).

None.

6. **Organizational Victims and Bankruptcy Cases.** Provide any information required under Fed. R. App. P. 26.1(b) (organizational victims in criminal cases) and 26.1(c) (bankruptcy case debtors and trustees). *See* Fed. Cir. R. 47.4(a)(6).

None/Not Applicable.

Dated: February 29, 2024

By: /s/ Meredith Martin Addy
Meredith Martin Addy

TABLE OF CONTENTS

CONFIDENTIAL MATERIAL OMITTED

The confidential information on pages 9, 20, 22-24 are words that identify code files from Apple technical documentation about its software that Apple has asserted are confidential and covered by the district court protective order.

TABLE OF AUTHORITIES

Cases

Amdocs (Isr.), Ltd. v. Openet Telecom, Inc.,
761 F.3d 1329 (Fed. Cir. 2014) 3, 17, 27

Enzo Biochem v. Gen-Probe, Inc.,
424 F.3d 1276, 1284 (Fed. Cir. 2005) 16

E-Pass Techs. Inc. v. 3COM Corp.,
473 F.3d 1213 (Fed. Cir. 2007) 17

Griffin v. United Parcel Serv., Inc.,
661 F.3d 216 (5th Cir. 2011) 15

Intellectual Sci. and Tech., Inc. v. Sony Electronics, Inc.,
589 F.3d 1179 (Fed. Cir. 2009) 17, 18

Liquid Dynamic Corp. v. Vaughan Co.,
449 F.3d 1209 (Fed. Cir. 2006) 19

Packet Intelligence LLC v. NetScout Sys.,
965 F.3d 1299 (Fed. Cir. 2020) 12, 17

Shackelford v. DeLoitte & Touche, LLP,
190 F.3d 398 (5th Cir. 1999) 19

UltimatePointer, LLC v. Nintendo Co.,
816 F.3d 816 (Fed. Cir. 2016) 14, 15, 25

Versata Software, Inc. v. Sap Am., Inc.,
717 F.3d 1255 (Fed. Cir. 2013) 3, 12, 17, 27

Vita-mix Corp. v. Basic Holding, Inc.,
581 F.3d 1317 (Fed. Cir. 2009) 19

ARGUMENT

I. Fintiv Never Asserted that Evidence of a "Widget" Was Unnecessary as Apple Blatantly Mischaracterizes and Selectively Quotes.

Apple mischaracterizes and selectively quotes Fintiv's opening brief to suggest Fintiv "argues that it need not offer any evidence of a widget in order to show infringement." RB33. Apple improperly contends that Fintiv's briefing "criticiz[ed] the alleged 'demand' that Fintiv present 'evidence of specific "widget" software code,'" RB37 (quoting OB24), but Apple conveniently leaves out the main point of that quote. Fintiv did not criticize the requirement to present evidence of "widget" software code, Fintiv's actual full statement was that Apple improperly demanded "Fintiv present ***direct*** evidence of specific 'widget' software code or source code." OB24 (emphasis added). Despite Apple's repeated mischaracterizations throughout its response, *see* RB1, RB29, RB33, RB37, Fintiv has never argued that it did not have to present evidence of widget software.

Rather, Fintiv has always accepted the burden to demonstrate infringement, including the burden to demonstrate the accused products act upon a "widget" as the claims require. OB32. The district court, however, erred by seeming to demand specific code locations and citations not required by the claims and focusing too heavily on a failure to identify specific widget software code in Apple's accused product. *See* OB26-27, OB30-34 (citing Appx00006-00007, Appx000010, Appx27254, Appx27274). This was problematic for several reasons.

First, the claims require software actions or steps that *are performed on* a widget. OB28-32; *see e.g.,* Appx098, 13:15-29 (Claim 11, "retrieving a widget"); Appx0098, 14:7-23 (Claim 18, "filter a widget"); Appx0098, 14:36-53 (Claim 23, "provision the … widget …."). Thus, Apple infringes if the accused products include software for acting upon widgets—regardless of whether Apple makes the widget itself. And Fintiv's expert report shows the presence of software for acting on a widget. *See, e.g.,* OB30-31, OB47-49; Appx18767; Appx18855-18856; Appx18860; Appx18905. Ignoring this evidence, Apple and the district court criticize Dr. Shamos for saying "I don't know," in response to a *different* question: whether the software Dr. Shamos identified as satisfying the claim language actually "is the widget." RB57 (citing Appx18967-18968); Appx00006-00007; Appx00008 n.1; Appx00010.

Second, as Apple admits, it does not make the widget code; rather, it receives that from third parties. Appx01595-01596; *see also* RB14 (noting interaction of system with third parties).[1] While the widget is ultimately implemented in Apple's

[1] Apple contends that its admissions that widgets come from third party sources early in the litigation was simply a misstatement, RB43, but Apple offers no evidence of its own to establish the contrary.

Wallet product as identified by Dr. Shamos and Fintiv's counsel, OB36-42, the Apple source code that acts upon the widget does not necessarily contain the widget code itself. Thus, not only is it perfectly reasonable not to find widget code in Apple's software, it is also perfectly reasonable for patent owners to rely on circumstantial evidence and expert analysis to identify what that software is acting upon—*i.e.,* a widget. *See Versata Software, Inc. v. Sap Am., Inc.*, 717 F.3d 1255, 1261 (Fed. Cir. 2013); *Amdocs (Isr.), Ltd. v. Openet Telecom, Inc.*, 761 F.3d 1329, 1342-43 (Fed. Cir. 2014). Just as a party may point to brochures or demonstration of a program acting on photos (rather than the digital ones and zeros that make up the photo's code) to establish a claim limitation requiring a code step for retrieving photos, OB32, it is reasonable to expect Fintiv to establish evidence that widgets are implemented by the accused software with expert analysis and demonstrations of Apple products implementing the widget rather than the widget code itself—especially where access to that code is limited.

Third, the specific location of widget code or its existence as a single file, is not required by the district court's construction of "widget." Indeed, the district court applied the broad, plain and ordinary meaning of "widget" as used in the art, that only required a widget that was "software that is either an application or works with an application, and which may have a user interface." Appx00067.

Although Apple contends that the facts and patent claims of this case are comparable to the "Downloadable" claim limitations in *Finjan LLC v. SonicWall, Inc.*, 84 F.4th 963, 970-71 (Fed. Cir. 2023), RB38-39; unlike *Finjan*, the parties here have an actual dispute over how the accused product works and the district court here rejected a claim construction similar to the one applied in *Finjan*.

In *Finjan*, the court considered claims for "receiving a Downloadable" and other actions. *Id.* at 970-71. But, the parties agreed that a "Downloadable" had to be "an *executable* application program, which is downloaded from a source computer and *run on the destination computer*." *Id.* at 971 (emphasis-in-original). The defendant submitted unrefuted evidence that the products operated by inspecting the payloads of packets on an individual basis and sent each packet to its destination without extracting data or reassembling the packet into an executable file. *Id.* Because the accused product never assembled the packets into an "executable form, *i.e.,* one that can be executed and run," the accused product never actually received a "Downloadable." *Id.*

This case is far different than *Finjan*. First, there is a factual dispute whether the Apple products operate on widget software. Unlike *Finjan*, Fintiv presented expert testimony explaining how the functionality presented in the product *must be* performed by executable software code and presented evidence from screenshots of the accused products *executing* those functions. *See* OB37-42 (identifying evidence). In fact, Apple presents no evidence—let alone unrefuted evidence as in *Finjan*—to explain how the functionality Dr. Shamos identified operates—other than through an executable widget code. *See, e.g.,* RB63-68. Second, the claim construction in *Finjan* required a "Downloadable" to be a complete executable program, whereas here, the district court rejected Apple's attempts to limit a "widget" to a complete application. Appx00066. Thus, there was never a need to show a single source code file that was the "widget" or demonstrate a complete program that was the "widget," like the "Downloadables" in *Finjan*. Software that "worked with an application" was sufficient. Appx00066.

II. Fintiv Presented Sufficient Evidence that the Accused Products Used "Widgets," Including Evidence the "Widget" Is Software That Works Within an Application and has a User Interface

During claim construction, the district court rejected Apple's proposed construction that would have required the "widget" to be a standalone software application, Appx00066, and instead adopted a plain and ordinary meaning of the "widget" term. Recognizing that "widget" had a broad meaning (*e.g.*, gizmo) and a narrower meaning for software, the district court clarified that the plain and ordinary meaning of a "widget" for the software arts was "software that is either an application or works with an application, and which may have a user interface," and further noted widgets could "reside in another application" and simply be "code … that runs within an application." Appx00063-00064; Appx00066-00067.

Using this construction, Fintiv collected evidence to establish that Apple's accused products use widgets including, expert analysis of the accused product operation that identified the functionality in the product that must be performed by widget software, OB36-42, and Apple's own documentation and deposition testimony that identified the use of widgets in the accused products, OB43-46. This evidence by itself, and as a whole, is sufficient to establish the use of "widgets" in Apple products.

A. Dr. Shamos's Expert Analysis Identified the Widget in Apple Products and Explained How He Knew it Was Software Working With An Application.

1. Dr. Shamos Did Not Admit There Was No "Widget" Code

Apple and the district court put considerable focus on the fact that Dr. Shamos could not determine whether the source code files specifically listed in his report were themselves the "widget" or part of it. *E.g.,* RB30, RB54, RB56-57; Appx00006-00010; OB25-27. While Apple attempts to twist this fact into an admission of non-infringement, that emphasis completely disregards Dr. Shamos's actual testimony and the context of the citations. The claim limitations do not simply claim a "widget." OB28-31. In every instance they require code that performs an action upon a widget (*e.g.*, retrieves, manages, etc.). Thus, as part of his expert analysis, Dr. Shamos identified evidence and code that performed that action and identified where in that product the widget existed and could be found. Dr. Shamos never claimed the source code files he identified were the complete "widget," nor did he have to do so.

For example, the often-cited paragraph 309 of Dr. Shamos's report asserted the basis for infringement of claim limitation 11[f] that required a method of "retrieving a widget." Appx18766-18768. Accordingly, Dr. Shamos identified the Apple source code responsible for retrieving widgets. Dr. Shamos was not required to identify the "widget" in those source code files and never asserted those files were the "widget." Instead, Dr. Shamos identified the widget from the functionality demonstrated and observed in the accused products—*i.e.*, "[t]he software (the 'widget') [that] allows a user to, for example, view the card's details or perform transactions." Appx18767; Appx18984-18985 (discussing ¶309).

Throughout Dr. Shamos's report he identified the evidence and Apple code for performing acts upon a "widget" and identified the specific feature of the product that demonstrated a "widget." OB37-42 (citing Dr. Shamos testimony and report). In multiple parts of his report and for multiple Apple products (*e.g.*, iOS, watchOS, etc.), he demonstrated where the widgets were from screenshots showing how users could select card images within the Wallet, which would activate widget code that then presented certain actions such as performing transactions, displaying card details, and other things. Appx18790-18791 ¶¶359-360; Appx18882-18884 ¶¶578-580; Appx18927-18928 ¶¶696-698.

Even now, Apple continues to assert that summary judgment should be granted because "[Dr. Shamos] agreed that none of [the source code files] ***is*** the claimed widget," RB56-57 (emphasis added), but this continued demand is erroneous for two reasons. First, nothing in the "widget" term or claim construction of "widget" requires there to be a single source code file that is the widget. Indeed, the district court's claim construction indicates the widget can simply "work with an application" and "provisioning" of a widget as recited in Claims 11 and 23 simply meant "making [the widget] available for use." Appx00067; Appx00083-00084 (construing "provisioning"). The "widget" construction, here, does not state how the widget code is provided or where it is made available for use. It only requires the "widget" to be "software that …works with an application." Appx00066-67.

Second, as Dr. Shamos explained, after agreeing the source code files were not the entire widget, "the widget is software that enable you to do things with the card" and "there is no concept of this thing called a widget that -- that everything is in. They're different – they're different components." Appx18961, Appx18965-18966.

2. Dr. Shamos Explained the Source Code Files and the Reason Why He Knows Apple's Products Use Widgets

Apple further suggests that Dr. Shamos's analysis is inadequate because he "never explained what any of the various source code files do, what they are acting on, or why a skilled artisan would understand that the thing they act on is software that qualifies as a 'widget.'" RB58. But all three of Apple's claims are false.

First, as Apple was even forced to admit, RB58-59, Dr. Shamos identified what multiple files did in relation to the widget. For example, he noted:

- Code ID – used to get the widget onto the machine (*i.e.,* retrieving).
- Code ID – involved in creating the widget.
- Code ID – configured to provision a widget.

See OB43; Appx18967-18968; Appx18905.

Second, Dr. Shamos identified upon what the files were operating. For example, he identified the widget as the "software…with a user interface" that is associated with the cards in the Wallet Application and allows users to view card details and perform transactions. Appx18767. He went further to provide demonstrative screenshots of the widgets he was describing and their ability to identify a screen tap and perform a function such as displaying card information or performing a transaction. Appx18790-18791 ¶¶359-360; Appx18794-18795 ¶¶368-369; *see also* Appx18882-18884 ¶¶578-579; Appx18885-18886 ¶¶584-586.

Third, Dr. Shamos described why a person would understand the widget he identified was software. For example, he explained that the demonstrative screenshots included a user interface "that is made available to the user for selecting, via its user interface, … to, for example view the card's details or perform transactions." Appx18790. He further explained how he could identify the widget as software because it allowed him to do things with the card, as opposed to simply view data as Apple's attorney suggested.[2]

> For example, I can select it to -- as a payment instrument or I can select it and view its details or I can select it and I can delete it. ***And so the interactivity that occurs between the user and that card information is performed by a widget***.
>
> * * *
>
> So the -- the ***widget is software that enables you to do things with the card.***

Appx18961 (emphasis added).

> So the widget consists of a bunch of things. One is that there is a – a screen manager and it manages objects on the screen. One of those objects that's on the screen is a – is card art that corresponds to a particular payment instrument.
>
> And ***the interface knows that when you touch within the area of that – that card art, code is invoked but then acquires the device – the payment instrument data for display on the – for display on the screen***.
>
> So that code is a combination of pre-existing Apple operating system code that has been specially configured during provisioning so that it understands this particular – where to get the data for this particular payment instrument.

Appx18963-18964 (emphasis added).

> So the widget is software. Software can have multiple components, some of which call each other, et cetera. And so there is

[2] *See* RB64 (contending without explanation the functions identified by Dr. Shamos could be performed with data alone).

> no concept of this thing called a widget that -- that everything is in. They're different – they're different components.
>
> The -- the code that recognizes the geographic coordinates of a touch on the screen is pre-existing in the Apple device because then that's how you interact with the Apple device is by -- is by touching the screen.
>
> Now, ***what happens when you touch a particular place, that changes depending on which widgets have been installed. That is, how the interface has been configured to deal with a particular card***.
>
> Before I provision a card there is nothing on the screen I can touch to get access to that card. ***Once I have provisioned a card, then the icon exists and the underlying code exists that retrieves the credentials associated with that card***.

Appx18965-18966 (emphasis added).

> [T]he widget is the code that's sitting behind the -- the card art that is activated when I touch it.

Appx15247.

Apple goes on to complain that Dr. Shamos further did not know whether the card art contained executable software. RB62. But again, this ignores the entirety of his testimony and is irrelevant to the infringement determination. During his deposition, Dr. Shamos was asked to confirm "the card art is not the widget itself." Appx18960; *see also* Appx15247-15248. He responded that there were ways the card art could be the widget (*e.g.,* if it was a PNG or PDF with executable code) but he noted that, even if the card art itself was not software, it would still be "a part of the user interface to executable code which causes various things to happen." Appx18961. In other words, regardless of whether the card art itself is "part of the widget" and "has executable code in it," it must still be a "gateway to the widget" with executable code to cause various things to happen. Appx15247-15248. It does not matter if the code is embedded in the card art or the code sits behind the art, in either case the functionality Dr. Shamos observed indicates there is widget software executing in the product.

B. Dr. Shamos and Fintiv Properly Relied on Product Functionality to Prove Infringement

Contrary to Apple's arguments, RB63-68, demonstrating the existence of a widget through software functionality, is not improper. Indeed, this Court has repeatedly approved expert demonstration and analysis of a product's "inherent functionality" as some of the "most telling evidence" to overcome summary judgment. *Versata*, 717 F.3d at 1261-62; *see also Packet Intelligence LLC v. NetScout Sys*., 965 F.3d 1299, 1306 (Fed. Cir. 2020) (confirming infringement from expert demonstration of "example"). In *Packet Intelligence*, for example, the expert provided an example of how the accused products "use the information in memory to create a 'key performance index' in a … white paper" and testified "that the feature 'demonstrate[d] that information in the flow record is sufficient to identify the flow-entry and also to allow it to associate with … conversation flows" as required by the claims. *Id.*

Here, Apple incorrectly contends Dr. Shamos assumed there was software that performed a functionality rather than prove the functionality was performed by software. RB64. But this is not true. Dr. Shamos's analysis of product functionality is the same as the analysis approved in *Packet Intelligence*. He explained from the screenshot examples and analysis of the product that "the widget is software that enables you to do things with the card," in particular, the interactivity with the card. Appx18961. As he explained, the Apple screen manager "manages objects on the screen" and "one of those objects that's on the screen … is card art that corresponds to a particular payment instrument." Appx 18963. As Dr. Shamos indicated, the interface knows that when a user "touch[es] within the area of that … card art, code is invoked but then acquires … the payment instrument data for … display on the screen." Appx. 18963-18964.

Apple contends that Dr. Shamos backtracked from the statement that the "screen manager" is in the widget, RB64, but this is also not true. Dr. Shamos clearly indicated that the screen manager was part of the Apple code. Appx18964. He specified that the "widget is software and it has to perform a certain function" and he specifically noted the function it was performing "is allowing the user to access his payment instrument." Appx18964-18965. He explained that while "the code that recognizes the geographic coordinates of a touch on the screen is pre-existing in the Apple device" (*i.e.,* the screen manager), when a user "touch[es] a particular place, that changes depending on which widgets have been installed." Appx18966. He went on to further explain how he knows the functionality is installing widget software, noting "the widget is the code that's sitting behind the -- the card art that is activated when I touch it" and prior to provisioning there is "nothing on the screen I can touch to get access to that card," but once provisioned "the icon exists and the underlying code exists that retrieves the credentials associated with that card." Appx18965-18966; Appx15247. Thus, the software widget code activates certain functions when the screen is touched, such as viewing certain information or performing transactions depending on the widget code installed. OB37-42.

Dr. Shamos's analysis of widget functionality is entirely different from the functionality analysis rejected in *UltimatePointer*. *See UltimatePointer, LLC v. Nintendo Co.*, 816 F.3d 816 (Fed. Cir. 2016). In that case, there was no dispute how the accused product worked. The court construed the term "handheld device" to be a "handheld direct pointing device" and held that the Wii remote accused in the litigation was not a direct pointing device because it used a sensor bar to determine the aim of the remote and not the direct interaction of the remote with the screen. *Id.* at 824. There, the experts did not contest the operation of the accused product and there was no dispute that the remote interacted with the sensor bar rather than directly with the screen. *Id.* at 825. While the patent owner contended that manuals and videos of the accused product suggesting a setup caused the system cursor to be displayed at the point of aim, that evidence only demonstrated how the remote was used and not how it worked. *Id.* at 824-25. As the court correctly pointed out, the operation of the remote "creat[ed] the illusion of direct pointing" device and not an actual direct pointing device. *Id.* at 825.

In *UltimatePointer,* the experts agreed that the sensor bar was responsible for the cursor position and the direct pointing functionality was merely an illusion, whereas here, Dr. Shamos's functionality analysis stands alone and is entitled to "all reasonable inferences." *See Griffin v. United Parcel Serv., Inc.*, 661 F.3d 216, 221 (5th Cir. 2011). In his analysis, Dr. Shamos explained that the functionality he reviewed must be performed by a widget and, more specifically, software interacting with the interface to display information and perform transactions. Appx18961-18966; Appx15247; Appx27375.

Though Apple suggests that the functionality may be performed in some other way using data, RB64, Apple presents no alternative evidence or explanation for how simple data would produce the functionality identified and certainly not any agreement between the parties on how that functionality is performed, if not by widget software. Apple's only evidence appears to be its continued reliance on admissions that the source code files Dr. Shamos identified were not the widget. RB54, RB57-58. But, as Dr. Shamos repeatedly explained, simply because the files were not the widget does not mean he could not identify the widget by the way the accused product functioned. Appx18961; Appx18963-18966; Appx15247; Appx27375. Such testimony is sufficient to survive summary judgment, especially when—as here—that testimony is left unrebutted in the record. *Cf. Enzo Biochem v. Gen-Probe, Inc.*, 424 F.3d 1276, 1284 (Fed. Cir. 2005) (holding "[a]ttorney argument is no substitute for evidence" when a party presents evidence to establish its burden, but the other side only presents "arguments and perhaps a suggestion of what [it] might present at trial").

Apple claims that the district court rejected functionality arguments during claim construction and that because the claim construction ruling is undisputed, Fintiv cannot assert functionality to prove infringement. RB32-33. But this is incorrect and makes no legal sense. Fintiv asserted that "widget" could be construed to have its plain-and-ordinary meaning, but to the extent it was defined, asserted a construction for "widget" that required "integrated functionality that relates to applications related to a financial institution, transportation account, and the like." Appx00063. The district court rejected the "functionality" construction because it might be vague for a jury to understand, not because it was not a valid way to establish infringement. Appx00064.

Rejecting a "functionality" term from a claim construction because it would not be helpful to the jury is not the same as proving infringement by analyzing the accused product functionality and explaining how that functionality demonstrates the specific claim limitations are met by the accused product. This Court repeatedly holds that infringement can be proven by examination of an accused product's functionality. *See Packet Intelligence*, 965 F.3d at 1306; *Versata*, 717 F.3d at 1261; *see also Intellectual Sci. and Tech., Inc. v. Sony Electronics, Inc.*, 589 F.3d 1179, 1185 (Fed. Cir. 2009) (noting that a party must identify "the reason one of skill in the art would recognize [feature] as infringing").

C. Dr. Shamos's Analysis Is Not Based on Unspecific or Speculative Evidence

Despite Apple's efforts, this is not a case that turns on unspecific circumstantial information or speculation. Apple points to Dr. Shamos's uncertainty regarding certain details of the product to suggest that Dr. Shamos is simply speculating about the existence of a widget. RB57-59; RB63-65. But uncertainty about where exactly software code for a widget exists in an accused product or uncertainty about every factual detail concerning how that software is implemented is irrelevant unless such detail is required by the claim. *See Amdocs*, 761 F.3d at 1343 (reversing non-infringement finding based on where location records were generated when the "generate" location was not a required claim limitation). For example, Apple's argument that Dr. Shamos did not know whether the card art existed in an executable PDF or PNG file is only relevant if the claims require the "widget" to be implemented by an executable card art file, which they do not.

This general premise is supported by the cases Apple cites to suggest that Fintiv's evidence is speculative. In *E-Pass Technologies*, for example, the patent claim was directed to a method that was only infringed when the steps were performed in an exact order. *E-Pass Techs. Inc. v. 3COM Corp.*, 473 F.3d 1213 (Fed. Cir. 2007). While the plaintiff presented evidence of accused product demonstrations and product manuals that "taught their customers each step of the claimed method in isolation," there was no evidence that taught "all of the steps of the claimed method together, much less in order." *Id.* at 1222. There is no such detailed requirement of the "widget" limitation in this case. Here, Fintiv was only required to demonstrate the Apple code was designed to act on a "widget", *i.e.*, "software that … works with an application, and which may have a user interface." Appx00067. Fintiv did not have to show a widget in a specific source code file or embedded within the image of a card. Rather, demonstrating the functionality of the accused product that was the widget from various screenshots and documents, then explaining why such functionality was widget software, as Dr. Shamos did, is far from mere speculation that a widget is used in Apple's products.

Apple's reliance on *Intellectual Science and Technology* is equally misplaced. In that case, the expert made conclusory assertions that the claim elements were met without pointing to the specific structure. 589 F.3d at 1184. While the patent owner identified schematics generally with a multitude of symbols, it failed to identify how the symbols indicated an infringing device. *Id.* Further, while the expert identified some structures as a claimed "ITDM," he failed to identify how that structure met the same level of output complexity required by the patent. *Id.* at 1185.

In this case, however, Dr. Shamos specifically identified what he claimed to be a widget—it is the software underlying the card images that allowed users to perform functions such as display information upon selection. *E.g.,* Appx18767. Also, unlike *Intellectual Sciences*, Dr. Shamos explained why the functionality identified must be software. *E.g.,* Appx18961-18966; Appx27375. The Apple widget infringes if it is simply software. There are no complex structures in the current claims that need to be proven as there were in *Intellectual Science*.

D. The Combination of "Widget" Evidence Creates a Genuine Issue of Fact

Unlike many of the cases Apple cites, Dr. Shamos's testimony does not stand alone. Indeed, his testimony must be judged among the totality of evidence presented, and a combination of multiple pieces of evidence—even if circumstantial—can establish a genuine issue of material fact that precludes summary judgment. *See Vita-mix Corp. v. Basic Holding, Inc.*, 581 F.3d 1317, 1326 (Fed. Cir. 2009) (combining circumstantial evidence testimony from multiple witnesses to establish sufficient evidence of infringement to survive summary judgment); *Liquid Dynamic Corp. v. Vaughan Co.,* 449 F.3d 1209, 1220 (Fed. Cir. 2006) (combining circumstantial evidence from manuals and engineering records to create inference of infringement); *Shackelford v. DeLoitte & Touche, LLP*, 190 F.3d 398, 409 (5th Cir. 1999) (holding the "totality of … evidence is sufficient to support

[an] inference" and that "the combination of suspicious timing with other significant evidence of pretext, can be sufficient to survive summary judgment").

To borrow Apple's analogy, this is not a case about seeing a picture of a vehicle and assuming it has an internal combustion engine. RB65. This is a case about seeing the picture of a vehicle with an exhaust pipe, labeling that says it only takes "unleaded", and an indication it is powered by a V8, 5.2-Liter engine that an automobile expert indicates are specifications for an engine that burns gasoline. While there may not be direct photographic evidence of gasoline burning in the engine cylinders, there is still more than enough evidence for a reasonable jury to conclude that the vehicle has an internal combustion engine. Despite Apple's attempts to disparage the individual pieces of Fintiv's evidence, their combination in addition to the proper inferences in Fintiv's favor, establish a genuine issue of material fact.

First, the evidence established that "passes" used in Apple software were related to and included "widgets." Apple does not even attempt to defend the district court's obvious misinterpretation of Dr. Shamos's testimony that concluded "passes are not the widget." Appx00010; RB69-70. As noted in its opening brief, Dr. Shamos's testimony cited by the district court for making this inference, never said that "passes are not the widget." OB52-54. While Dr. Shamos, could not determine if specific source code files, such as "Code ID" and "Code ID" were the actual widget source code or part of them, Appx18963, Appx18968, he never said or indicated that a pass is not a widget or contain widgets. Indeed, he noted that those specific files were involved in the creation of widgets. Appx18968.

This is important because the district court's legally-flawed interpretation of Dr. Shamos's testimony to conclude "passes are not the widget," Appx00010, allows it to then ignore straightforward inferences from the evidence that widgets are related to passes and contain software regardless of the name Apple assigns. For example, Apple readily admits that "pass packages" are received by the accused products and that "pass" contains card image files and metadata. RB15. These are some of the same features Dr. Shamos pointed to in identifying the "widget." Appx18767 ¶309 ("the widgets can be retrieved, including their associated software code such as card image code and metadata code"). However, while Apple suggests passes are just card images and data, RB15, that assertion ignores the testimony of its own witness that further acknowledges those "passes" contain "a [user interface] and *software presentation* on the application processor that relates to the card that was installed." Appx19378 (emphasis added).

This testimony and Apple's admission that "passes" contain card art and data confirms that "passes" as used in Apple's accused products and files directly relate to the cards, card art, user interface, and software presentation that Dr. Shamos identified as the "widget" from the functionality he observed. OB37-42. In other words, Apple's witnesses confirm the same software functionality in a "pass" that Dr. Shamos identified as the "widget." The district court clearly erred when, despite Apple's own witness admissions to the contrary, the court inferred in Apple's favor that because Dr. Shamos admitted the "[redacted]" and "[redacted]" files were not the widgets, then passes generally are not the widget.

Apple contends there is nothing to connect a "pass" to a "widget," RB70, but this is simply wishful thinking that ignores the admissions of its own witnesses and the reasonable inferences a court must draw from that evidence at summary judgment. In fact, further evidence connects passes and widgets. Aside from Dr. Shamos's testimony that "[redacted]"-labeled source code files are involved in creating and performing other actions upon a widget, *see* OB43; Appx18967-18968; Appx18905, at least one source code file for Apple's Wallet product named "[redacted]" directly links [redacted] and [redacted] within the code for the accused products.

Apple criticizes the [redacted] file because Dr. Shamos only testified it "appears to be related to viewing the widgets" and did not say it ***was*** the "widget," RB70-71, but this argument misses the point. If the [redacted] file was the only evidence Fintiv presented that Apple's accused products use "widgets," then Apple would probably be correct that there should be more about what that file does and how it is connected to the claims.

Confidential Material Omitted

However, the Code ID file does not stand alone. It stands with expert testimony that the Code ID-labeled source code files and others are used by the accused products to act on widgets, as well as expert testimony that the functionality seen in the accused product must be provided by widgets. *See* OB43; Appx18961-18968; Appx18905; Appx15247; Appx27375. It further stands with Apple witness testimony that indicates "passes" include user interfaces, a software presentation, and card images, which is virtually identical to the qualities of a widget supplied by the district court's construction and Dr. Shamos's analysis. Appx19378. In light of the cumulative evidence regarding similar pass and widget functionality admitted by Apple, the reasonable inference to be made from the terminology used in Code ID file, (emphasis added) is that it confirms simply that "widgets" are related to passes and used with the passes and Code ID-labeled files of the accused product as Dr. Shamos identified.

Apple argues that the Code ID file cannot be linked to the accused products because it is not used in iOS devices, RB71, but offers no evidence of that. Instead, its own witness explained that Code ID in the filename gave "a clue that this is some part of the user interface" and that line 3 of the file used the term Code ID that was "framework in the Wallet project that's used exclusively on iOS devices, notably the iPhone." Appx19366. Apple's response seems to suggest that it could be inferred that only Code ID was in the iOS code, but even if Apple does try to make that inference at trial, the inference applied at summary judgment should be for Fintiv. That is, an inference the file is used exactly where the witness suggested it would be—the iOS framework. In fact, counsel's suggestion that only [redacted] and not the [redacted] file was used on iOS is directly undermined by the next statement in Apple's witness's testimony that the [redacted] class is a subclass of [redacted] which is "part of the [redacted] framework that is for use on iOS devices like the iPhone." Appx19366.

Ultimately, Apple cannot make inferences both ways to avoid infringement. On the one hand, Apple argues that the code is in the MacOS but not used by MacOS. RB71 (citing Appx00008 and Appx19365-19366). But, when describing how the code would be part of user interface, [redacted] and [redacted] framework for "the Wallet project that's used exclusively on iOS devices," Apple argues the file was only "Mac source code, not iOS source code." RB71. Apple cannot use a witness to disclaim the use of the file in Apple's MacOS because it claims the file is only used, in iOS and then suggest the file is not used in iOS because it is in MacOS. Fintiv is at least entitled to a reasonable inference one way or the other, and either way establishes sufficient evidence of widget use in the accused product.

Furthermore, Apple criticizes the evidence that it uses the term "widget" in other parts of its products. RB72-73. But again, while the use of the term "widget" to describe similar features in other parts of its products outside the accused product features may not be sufficient evidence standing alone, it is relevant to determine the meaning of "widget" applied by Apple. Though the district court construed "widget" with the plain and ordinary meaning of that term to persons in the software computer field, Apple's clear use of the term elsewhere in its products and similarity of those "widgets" to the functionality identified by Dr. Shamos in Apple's products and the use in relation to software, confirms that Apple's use of widget within the accused product features would more than likely have the same meaning.

E. Apple Fails to Set Forth Alternative Evidence

Unlike the cases Apple cites that grant summary judgment, Apple fails to supply its own evidence to explain how the product functionality identified by Dr. Shamos operates, if not with software. For example, in *Finjan* and *UltimatePointer,* the defendants set forth their own unrefuted expert testimony and evidence regarding how the products functioned. *Finjan*, 84 F.4th at 971; *UltimatePointer,* 816 F.3d at 825. The summary judgment record here, however, is devoid of any such counter-evidence to explain the functionality Dr. Shamos identified. Rather, the sole evidence is presented by Fintiv.

Apple claims, without support, that the screenshots produced by Dr. Shamos could be generated by data alone rather than widget software. RB64. However, Apple never contradicts Dr. Shamos's explanation of the interactivity of the device card images with users, as they select cards to identify additional functionality such as card details, and to perform transactions.

Apple may be correct that there is no battle of the experts, but that is only because it brings no expert evidence to battle on Dr. Shamos's analysis. Ultimately, the accumulation of evidence presented by Fintiv is too great; and it is not sufficient for Apple to simply say there is not enough evidence of software without providing at least an alternative explanation for how the functionality works without software.

III. Apple's Arguments Demonstrate the Unavoidable Conclusion that the District Court Required Direct Evidence of Software or Source Code

Throughout its briefing, Apple repeatedly claims neither its arguments nor the district court demanded code, RB44-46, but its response brief tacitly acknowledges that Apple demanded direct evidence of software code of some sort. Apple repeatedly argues that whatever evidence Fintiv presents, it is not the actual widget software code, and therefore, Fintiv did not prove the widgets in the accused product were software. For example, Apple acknowledges that Dr. Shamos explained software was behind the card images in the accused products to make them act a certain way, display information, and perform transactions, but asserts summary judgment was properly granted because Dr. Shamos "admitted he did not find any such software in his investigation, and Fintiv still identifies none on appeal." RB31. Yet, the only thing Dr. Shamos did not identify was the specific widget source code, which the court's claim construction does not require. Appx00067.

Apple does nothing to clarify what, besides lines of code, Apple is criticizing Dr. Shamos for not presenting. He demonstrated where the accused product's widget could be found from at least the product demonstration and explained how he knew it was a widget with software. OB37-42. This is more than enough to survive summary judgment. *See Versata*, 717 F.3d at 1261-62. But Fintiv went even further to present evidence that Apple acknowledges the use of widgets in the accused product code, and Apple used functionality with the same widget definition applied by the court, and that functionality even Apple agreed were widgets. OB43-46, OB52-59. Apple offers no explanation for what precisely is missing besides source code.

In the end, Apple cannot have it both ways by suggesting that its arguments and the district court's grant of summary judgment did not demand some *direct evidence* of software and lines of code, but then criticize all other forms of evidence as failing to specifically identify the widget software. Apple's circular arguments must be rejected because they demand that ***only*** source code is good enough to establish software claim elements. Either direct evidence and location of specific software code is required to prove infringement or it is not, and here, this Court has definitively agreed that it is not required to establish infringement when other evidence exists. *Amdocs*, 761 F.3d at 1341.

CONCLUSION

This case should be reversed and remanded because the district court improperly required source code as evidence, failed to consider Fintiv's other relevant evidence of a "widget" in Apple's accused devices, and drew inferences in favor of Apple instead of Fintiv.

February 29, 2024

Jonathan K. Waldrop
Darcy L. Jones
Marcus A. Barber
John W. Downing
Heather S. Kim
ThucMinh Nguyen
Kasowitz Benson Torres LLP
333 Twin Dolphin Drive
Suite 200
Redwood Shores, CA 94065
650.453.5170
jwaldrop@kasowitz.com
djones@kasowitz.com
mbarber@kasowitz.com
jdowning@kasowitz.com
hkim@kasowitz.com
tnguyen@kasowtiz.com

Paul G. Williams
Kasowitz Benson Torres LLP
1230 Peachtree Street, NE
Suite 2445
Atlanta, GA 30309
404.260.6102
pwilliams@kasowitz.com

Respectfully submitted,

Meredith Martin Addy

Meredith Martin Addy
Charles A. Pannell, III
ADDYHART P.C.
10 Glenlake Parkway, Suite 130
Atlanta, Georgia 30328
312.320.4200
meredith@addyhart.com
cpannell@addyhart.com

Caren A. Yusem
ADDYHART P.C.
1101 Pennsylvania Avenue, N.W.
Suite 300
Washington, DC 20004
312.804.4885
caren@addyhart.com

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATIONS

Fintiv, Inc. v. Apple Inc.

23-2208

This brief complies with the relevant type-volume limitation of the Federal Rule of Appellate Procedure and Federal Circuit Rules because it has been prepared using a proportionally-spaced typeface and includes 6509 words.

Dated: February 29, 2024

By: /s/ *Meredith Martin Addy*
Meredith Martin Addy

UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

CERTIFICATE OF CONFIDENTIAL MATERIAL

Case Number: 22-2208

Short Case Caption: Fintiv, Inc. v. Apple, Inc.

Instructions: When computing a confidential word count, Fed. Cir. R. 25.1(d)(1)(C) applies the following exclusions:

- Only count each unique word or number once (repeated uses of the same word do not count more than once).
- For a responsive filing, do not count words marked confidential for the first time in the preceding filing.

The limitations of Fed. Cir. R. 25.1(d)(1) do not apply to appendices; attachments; exhibits; and addenda. *See* Fed. Cir. R. 25.1(d)(1)(D).

The foregoing document contains 12 number of unique words (including numbers) marked confidential.

☑ This number does not exceed the maximum of 15 words permitted by Fed. Cir. R. 25.1(d)(1)(A).

□ This number does not exceed the maximum of 50 words permitted by Fed. Cir. R. 25.1(d)(1)(B) for cases under 19 U.S.C. § 1516a or 28 U.S.C. § 1491(b).

□ This number exceeds the maximum permitted by Federal Circuit Rule 25.1(d)(1), and the filing is accompanied by a motion to waive the confidentiality requirements.

Date: 2/29/2024

Signature: /s/ Meredith Martin Addy

Name: Meredith Martin Addy

CERTIFICATE OF SERVICE

I hereby certify that on March 20, 2024, I filed Appellant Fintiv, Inc.'s Corrected Non-Confidential Reply Brief with the Clerk of the United States Court of Appeals for the Federal Circuit via the CM/ECF system.

Meredith Martin Addy
Meredith Martin Addy